1. An application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions residing on a computer-readable medium, wherein the program can be loaded to and executed by a resource-constrained device that is based on a processor architecture of fewer than 32 bits.

2. The software program of claim 1 wherein the program can be executed by a resource-constrained device based on a 16-bit processor architecture.

3. The software program of claim 1 wherein the program can be executed by a resource-constrained device based on an 8-bit processor architecture.

4. The software program of claim 1 wherein each instruction includes an 8-bit operation code.

5. The software program of claim 1 wherein the sequence of instructions is hardware platform-independent.

6. The software program of claim 1 wherein the instructions were converted from at least one Java class file and wherein at least some references to a constant pool were transformed to inline data.

7. The software program of claim 6 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operands in at least some of the instructions.

8. The software program of claim 6 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operation codes in at least some of the instructions.

9. The software program of claim 1 wherein the instructions can be executed by a virtual machine running on a microprocessor residing on the resource-constrained device.

10. The software program of claim 1 wherein the instructions can be executed on a portable smart card.

11. The software program of claim 1 wherein the instructions can be executed by a device that supports multiple data types, wherein the sequence of instructions includes data manipulation instructions, and wherein each data manipulation instruction is specific to a particular data type.

12. The software program of claim 11 wherein the data type associated with each data manipulation instruction is selected from among one of the following types: an 8-bit signed two's complement integer numeric type, a 16-bit signed two's complement integer numeric type and a 32-bit signed two's complement integer numeric type.

13. The software program of claim 11 wherein the instructions can be executed by a device that supports multiple reference types and wherein each reference type is selected from among one of the following types: a class type, an interface type and an array type.

14. The software program of claim 1 wherein the program includes at least one composite instruction for performing an operation on a current object.

15. An application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions residing on a computer-readable medium, wherein the program can be loaded to and executed by a resource-constrained device having random access memory with a capacity of no more than about 64 kilo-bytes.

16. The software program of claim 15 wherein the program can be executed by a resource-constrained device having random access memory with a capacity of no more than about 4 kilo-bytes.

17. The software program of claim 15 wherein each instruction includes an 8-bit operation code.

18. The software program of claim 15 wherein the sequence of instructions is hardware platform-independent.

19. The software program of claim 15 wherein the instructions were converted from at least one Java class file and wherein at least some references to a constant pool were transformed to inline data.

20. The software program of claim 19 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operands in at least some of the instructions.

21. The software program of claim 19 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operation codes in at least some of the instructions.

22. The software program of claim 15 wherein the instructions can be executed by a virtual machine running on a microprocessor residing on the resource-constrained device.

23. The software program of claim 15 wherein the instructions can be executed on a portable smart card.

24. The software program of claim 15 wherein the instructions can be executed by a device that supports multiple data types, wherein the sequence of instructions includes data manipulation instructions, and wherein each data manipulation instruction is specific to a particular data type.

25. The software program of claim 24 wherein the data type associated with each data manipulation instruction is selected from among one of the following types: an 8-bit signed two's complement integer numeric type, a 16-bit signed two's complement integer numeric type and a 32-bit signed two's complement integer numeric type.

26. The software program of claim 24 wherein the instructions can be executed by a device that supports multiple reference types and wherein each reference type is selected from among one of the following types: a class type, an interface type and an array type.

27. The software program of claim 15 wherein the program includes at least one composite instruction for performing an operation on a current object.

28. A resource-constrained device comprising:

memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions;

random access memory having a capacity of no more than about 64 kilo-bytes; and

a virtual machine implemented on a microprocessor wherein the virtual machine is capable of executing the sequence of instructions.

29. The device of claim 28 wherein the microprocessor is based on an 8-bit architecture.

30. The device of claim 28 wherein the microprocessor is based on a 16-bit architecture.

31. The device of claim 28 wherein each instruction includes an 8-bit operation code.

32. The device of claim 28 wherein the sequence of instructions is hardware platform-independent.

33. The device of claim 28 wherein the instructions were converted from at least one Java class file and wherein at least some references to a constant pool are transformed to inline data.

34. The device of claim 33 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operands in at least some of the instructions.

35. The device of claim 33 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operation codes in at least some of the instructions.

36. The device of claim 28 wherein the virtual machine supports multiple data types, wherein the sequence of instructions includes data manipulation instructions, and wherein each data manipulation instruction is specific to a particular data type.

37. The device of claim 28 wherein the program includes at least one composite instruction for performing an operation on a current object.

38. A resource-constrained device comprising:

memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions; and

a virtual machine implemented on a microprocessor that is based on an architecture of less than 32 bits, wherein the virtual machine is capable of executing the sequence of instructions.

39. A resource-constrained device comprising:

memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions;

random access memory having a capacity of no more than about 64 kilo-bytes; and

a processor capable of executing the sequence of instructions.

40. The device of claim 39 wherein the processor is based on an 8-bit architecture.

41. The device of claim 39 wherein the processor is based on a 16-bit architecture.

42. A resource-constrained device comprising:

memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions;

random access memory having a capacity of less than about 64 kilo-bytes; and

an application-specific integrated circuit (ASIC) capable of executing the sequence of instructions.

43. The device of claim 42 wherein the ASIC is based on an 8-bit architecture.

44. The device of claim 42 wherein the ASIC is based on a 16-bit architecture.

45. A smart card comprising:

memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions; and

a virtual machine implemented on a microprocessor, wherein the virtual machine is capable of executing the sequence of instructions.

~~46. The smart card of claim 45 wherein the virtual machine is substantially a Java Card virtual machine.~~

~~47. The smart card of claim 45 wherein each instruction includes an 8-bit operation code.~~

~~48. The smart card of claim 45 wherein the sequence of instructions is hardware platform-independent.~~

~~49. The smart card of claim 45 wherein the instructions were converted from at least one Java class file and wherein at least some references to a constant pool are transformed to inline data.~~

~~50. The smart card of claim 45 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operands in at least some of the instructions.~~

~~51. The smart card of claim 45 wherein the instructions comprise operation codes and operands and wherein at least some references to the constant pool are inlined into operation codes in at least some of the instructions.~~

~~52. The smart card of claim 45 wherein the virtual machine supports multiple data types, wherein the sequence of instructions includes data manipulation instructions, and wherein each data manipulation instruction is specific to a particular data type.~~

~~53. The smart card of claim 45 wherein the program includes at least one composite instruction for performing an operation on a current object.~~

~~54. A method of using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the method comprising:~~

~~receiving the software program in a resource-constrained device having random access memory with a capacity of no more than about 64 kilo-bytes; and~~

~~executing the sequence of instructions on the resource-constrained device.~~

~~55. The method of claim 54 further including:~~

~~storing the sequence of instructions on the resource-constrained device.~~

~~56. The method of claim 54 further including accessing the software program over a computer network prior to downloading the program onto the resource-constrained device.~~

~~57. The method of claim 54 further including accessing the software program over the Internet prior to downloading the program onto the resource-constrained device.~~

~~58. The method of claim 54 further including:~~

~~transforming constant pool indices that appear in the received set of instructions to corresponding data values.~~

add a1